Numerical simulation of turbulent free-surface flow in offshore applications

Veldman, Arthur; Luppes, Roelf; van der Heiden, Hendrik; van der Plas, Peter; Helder, Joop; Bunnik, Tim

Published in:
ECCOMAS VI International Conference on Computational Methods in Marine Engineering MARINE 2015

VI International Conference on Computational Methods in Marine Engineering
MARINE 2015
F. Salvatore, R. Broglia and R. Muscari (Eds)

NUMERICAL SIMULATION OF TURBULENT FREE-SURFACE FLOW IN OFFSHORE APPLICATIONS

Arthur E.P. Veldman*, Roel Luppes*, Henri J.L. van der Heiden*,†, Peter van der Plas*, Tim Bunnik† and Joop Helder†

* Institute for Mathematics and Computer Science, University of Groningen,
P.O. Box 407, 9700 AK Groningen, The Netherlands
e-mail: {a.e.p.veldman, r.luppes, p.van.der.plas}@rug.nl
web page: http://www.math.rug.nl/∼veldman/comflow/comflow.html

† MARIN, P.O. Box 28, 6700 AA Wageningen, The Netherlands
e-mail: {h.v.d.heiden, t.bunnik, j.helder}@marin.nl

Key words: CFD; VOF-method; Run-up; Wave loading; Turbulence model

Abstract. To study extreme hydrodynamic wave impact in offshore and coastal engineering, the VOF-based CFD simulation tool ComFLOW is being developed. In this paper we will present its turbulence modeling. In particular, a blend of a QR-model and a regularization model has been designed. The QR-model belongs to a class of modern eddy-viscosity models, where the amount of turbulent eddy viscosity is kept minimal. Also, to enhance efficieny, local grid refinement has been added. For validation, experiments have been carried out at MARIN.

1 INTRODUCTION

Over the recent years, in close cooperation with the offshore industry, the VOF-based CFD simulation tool ComFLOW has been developed; see e.g. [1–3]. In the early phase of its development, emphasis has been on simulating momentum-dominated phenomena, such as the impact of extreme waves (green water loading [4]; wave run-up [5]) and sloshing (LNG tanks [6]; onboard spacecraft [7]). In these applications viscous effects can be mostly neglected. Later, the application area has been extended to flows where the influence of viscosity is becoming noticeable, like in side-by-side mooring or inside moonpools. One has to realize that, in general, the grids used by ComFLOW will be too coarse to resolve the turbulent details of the flow. Therefore a turbulence model is required that can cope with coarse grids, yet without excessive diffusive smoothing.

Numerical simulation of turbulent flow has to face the challenge of the very small spatial and temporal scales present in turbulence, requiring computational grids and time steps that resolve these small scales. Thus over the years strategies have been developed

to model the effects of the subgrid scales onto the resolved scales: turbulence modeling through RaNS or LES models. Most eddy-viscosity models (like k-ϵ or Smagorinski) add an excessive amount of 'turbulent' diffusion to model the dissipative effect of turbulence. Doing so, in laminar and transitional flow regions they seriously disturb the physical flow phenomena.

Therefore, in modern turbulence models the amount of turbulent diffusion is better controlled. One such model is the QR-model by Verstappen [8]. Based on functional-analytic arguments, it estimates the unresolved subgrid-scale details, and minimizes the amount of turbulent diffusion that is added. These estimates can be described in terms of the second and third invariants, $\mathcal{Q}$ and $\mathcal{R}$, of the rate-of-strain tensor. This method not only recognizes laminar parts of the flow, but also whether the turbulent flow field is more or less two-dimensional (relevant near free surfaces). In regions of backscatter, the QR-model is extended with a non-diffusive regularization model, which reduces the production of the smaller scales. Irregular geometries are treated with a cut-cell immersed-boundary method. Because in the engineering applications envisaged insufficient resolution in wall regions can be expected, a local refinement strategy and a wall model is employed [9].

The behaviour of the flow model will be demonstrated on simulations of the water motion inside moonpools, and on extreme wave impact against a semi-submersible offshore platform. For validation purposes, MARIN has performed a series of experiments.

2 THE NAVIER–STOKES EQUATIONS

Incompressible, turbulent fluid flow is modelled by means of the Navier–Stokes equations

$$\mathcal{M}u = 0, \qquad \frac{\partial u}{\partial t} + \mathcal{C}(u)u + \mathcal{G}p - \mathcal{D}u = f. \tag{1}$$

Here $\mathcal{M}$ is the divergence operator[1] which describes conservation of mass. Conservation of momentum is based on the convection operator $\mathcal{C}(u)v \equiv \nabla(u \otimes v)$, the pressure gradient operator $\mathcal{G} \equiv \nabla$ (absorbing the density ρ), the diffusion operator $\mathcal{D}(u) \equiv \nabla \cdot \nu \nabla u$ (with ν the kinematic viscosity) and the forcing term f.

The Navier–Stokes equations (Eq. 1) are discretized on a staggered Arakawa C-grid. The second-order finite-volume discretization of the continuity equation at the new time level $\cdot^{(n+1)}$ is given by

$$M^0 u_h^{(n+1)} = -M^\Gamma u_h^{(n+1)}, \tag{2}$$

where M^0 acts on the interior of the domain and M^Γ acts on the boundaries of the domain. In the discretized momentum equation, convection $C(u_h)$ and diffusion D are discretized explicitly in time. The pressure gradient is discretized at the new time level. In this exposition, for simplicity reasons the first-order forward Euler time integration will

[1]Note that calligraphic symbols denote analytic operators, whereas their discrete counterparts will be denoted by upper-case italic symbols.

be used. In the actual calculations, the second-order Adams–Bashforth method is being applied.

Using the diagonal matrix Ω to denote the control volumes, gives the discretized momentum equation as

$$\Omega \frac{u_h^{(n+1)} - u_h^{(n)}}{\delta t} = -C(u_h^{(n)})u_h^{(n)} + Du_h^{(n)} - Gp_h^{(n+1)} + f. \tag{3}$$

The discrete convection operator is skew-symmetric, i.e.

$$C(u_h^{(n)}) + C(u_h^{(n)})^T = 0, \tag{4}$$

where the superscript $\cdot^T$ denotes the transpose. In this way convection does not contribute to energy production or dissimination [10]: its discretization preserves the energy of the flow and does not produce artificial viscosity. This can be seen by looking at the discrete evolution of the energy $||u_h||_h$:

$$\begin{aligned} \frac{\partial}{\partial t}\tfrac{1}{2}||u_h||_h^2 &= (\!(\frac{\partial}{\partial t}u_h, u_h)\!)_h = (\!(-C(u_h)u_h - Gp_h + Du_h, u_h)\!)_h \\ &= -(\!(C(u_h)u_h, u_h)\!)_h - (\!(Gp_h, u_h)\!)_h + (\!(Du_h, u_h)\!)_h. \end{aligned} \tag{5}$$

When the operator $C(u_h)$ is skew-symmetric, the first term in the right-hand side of Eq. 5 vanishes, i.e. no numerical diffusion is introduced. To make the discretization fully energy-preserving, the discrete gradient operator and the divergence operator are each others negative transpose, i.e. $G = -M^{0T}$, thus mimicking analytic symmetry $\nabla = (-\nabla\cdot)^T$, as in [10,11]. In this way, also the work done by the pressure drops out of Eq. 5. In all cases, possible influence of boundary conditions is ignored.

The solution of the discrete Navier–Stokes equation is split into two steps. An auxiliary variable $\tilde{u}_h$ is defined through the equation

$$\Omega \frac{\tilde{u}_h - u_h^{(n)}}{\delta t} = -C(u_h^{(n)})u_h^{(n)} + Du_h^{(n)} + f. \tag{6}$$

Imposing discrete mass conservation (Eq. 2) at the new time level $(n+1)$ results in a linear system for the pressure:

$$\delta t\, M^0 \Omega^{-1} G\, p_h^{(n+1)} = M^0 \tilde{u}_h + M^\Gamma u_h^{(n+1)}. \tag{7}$$

This equation is often referred to as the discrete pressure Poisson equation, as it can be regarded to be a discretization of the equation $\mathcal{M}\mathcal{G}p = \mathcal{M}\tilde{u}$.

The liquid region and the free liquid surface is described by an improved VOF-method; for details we refer to [1, 12].

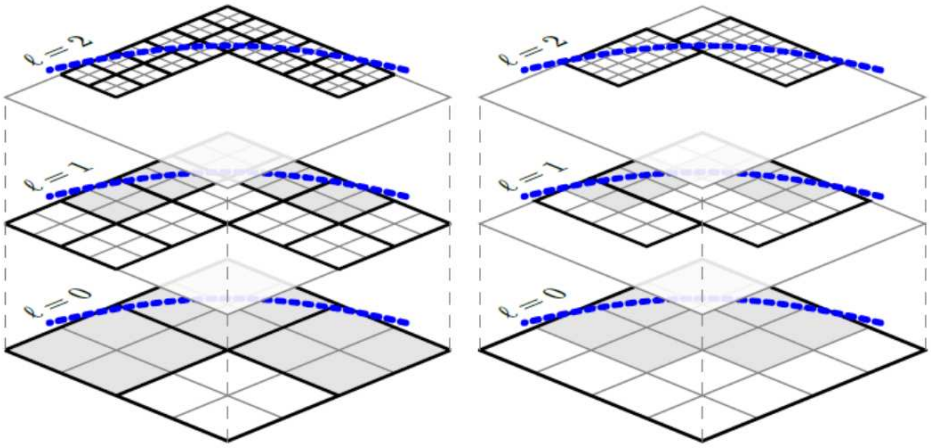


Figure 1: Block-based (*left*) and patch-based (*right*) refinement, illustrated for refinement around a curved strip.

3 LOCAL GRID REFINEMENT

Figure 1 illustrates the block-based and patch-based local refinement methods that are commonly encountered in the literature. In the present work, a block-based Cartesian local grid refinement approach is followed in which refinement and coarsening is applied block-wise. By setting the block size equal to 1×1, effectively a cell-based approach is obtained. On static grids, the blocks can be manually grouped together to form arbitrary rectangular refinement regions. On adaptive grids a block-clustering algorithm could be used to group blocks together in larger rectangular regions, but design of an efficient clustering algorithm is not a trivial task.

Discretization at interfaces As an example we consider refinement interfaces in the "$x =$ constant"-plane where the refined cells are located to the left of the interface. Five other interface orientations are possible, which are treated similarly. To further simplify discussion we assume a base grid with uniform grid spacings δx_0 and δy_0. In the current discussion we apply no refinement perpendicular to the refinement interface; a 1:2 local refinement is only applied along the interface. For the grid spacings this implies $\delta y_{\ell+1} = \delta y_\ell/2$ and $\delta x_{\ell+1} = \delta x_\ell$. Extending the discretization to the three-dimensional case, non-uniform grids and other refinement directions or refinement ratios is straightforward.

Typically, a large stencil is used for the approximation of missing pressure or velocity variables along the refinement interface. Interpolation of missing variables increases the number of non-zero coefficients in the pressure Poisson matrix, which might result in a non-symmetric matrix, putting higher demands on the solver. Most authors use a non-overlapping interface and apply linear (or even higher-order) interpolation for missing variables on the other side of the interface, e.g. [13]. Another approach is to apply linear

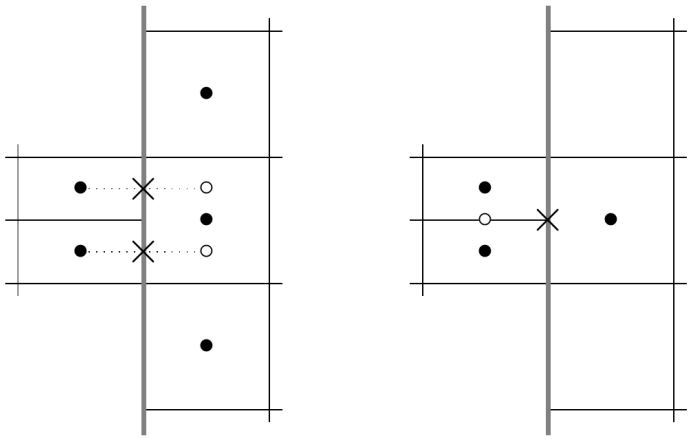


Figure 2: *Left*: Refinement approach with interpolation of missing pressure variables. *Right*: Refinement approach (as followed here) with shifted pressure gradient which is equal for both refined cell faces.

interpolation inside an overlapping interface [14]. In all these cases the discretization results in a non-symmetric system of equations.

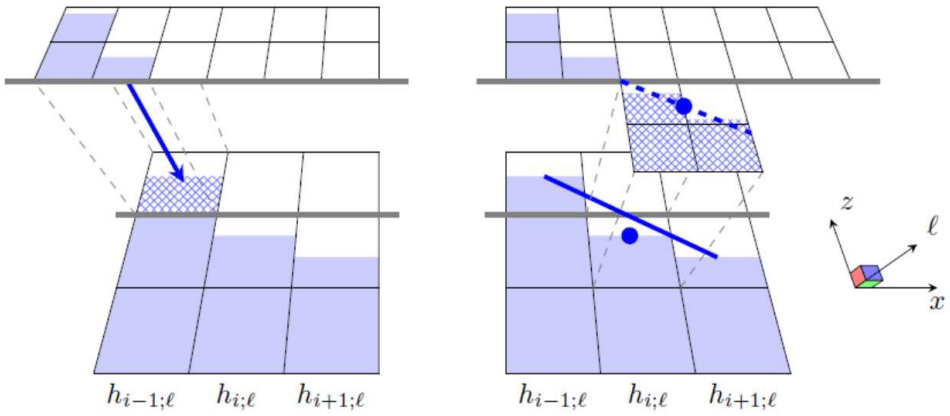
In the present approach, a compact discretization scheme is designed (in both space and time), which results in a small and symmetric scheme for the discrete composition of M and G. This makes it possible to employ an efficient linear solver. Furthermore, this facilitates the use of adjacent refinement regions as well as the interface discretization near objects and free-surface boundaries. The idea of using a compact discretization scheme that maintains the symmetry of the discrete Poisson equation can also be found in [15].

Divergence and pressure gradient There are two ways of obtaining a first-order accurate discretization of the pressure gradient. Either by using a linear interpolation for the missing pressure variable outside the refinement region (see left of Fig. 2) or by slightly shifting the location of the pressure gradient (see right of Fig. 2). Both approaches result in a first-order accurate discretization scheme introducing an error term that is proportional to respectively $\Delta\frac{\partial p}{\partial y}$ and $\Delta\frac{\partial^2 p}{\partial y \partial x}$, where for brevity we use Δ to denote the order of magnitude of a grid cell, omitting any subscripts or products of grid spacings. However, the first approach results in a relatively large stencil whereas the second approach uses a smaller interpolation stencil consisting of pressure variables that already form part of the regular stencil. For this reason the second approach is followed, which can be described as "using a constant pressure gradient along a refined cell face"; see e.g. [15, 16]. Correspondingly, we use a uniform velocity across the entire refined cell face and only place coarse computational velocity variables at the interface.

Convection and diffusion Missing velocities that are needed in the convection and diffusion scheme are approximated using (bi-)linear interpolation and a quadratic correction is added whenever sufficient information is available. Due to this interpolation, the

central discretization scheme for convection remains second-order accurate since plugging in an interpolation error ϵ of order $O(\Delta^3)$ in the numerator of a single differencing scheme results in a discretization error of order $\epsilon/\Delta = O(\Delta^2)$. The central discretization scheme for diffusion becomes first-order accurate since plugging in an interpolation error ϵ of order $O(\Delta^3)$ in the numerator of a double differencing scheme results in a discretization error of order $\epsilon/\Delta^2 = O(\Delta)$.

Locally refined Volume-of-Fluid (VOF) scheme The fluid distribution is discretized by means of cell-wise volume fractions which are advected by a second-order advection scheme. The fluid displacement in ComFLOW is modeled by means of a second-order Volume-of-Fluid advection scheme. A sharp free-surface interface is reconstructed from the volume fractions F by a piecewise-linear interface reconstruction (PLIC) developed by Youngs [17]; see also [18]. At refinement interfaces missing volume fractions



(a) Projection onto the parent grid (b) Reconstruction in the guard layer

Figure 3: Reconstruction of missing volume fractions needed by the VOF advection scheme.

on the coarse grid are approximated by averaging the refined volume fractions (left part of Fig. 3) and on the fine grid they are reconstructed geometrically from the volume fractions on the coarse grid (right part of Fig. 3). The advective fluxes located at the refinement interfaces are calculated on the fine grid and simply added up to obtain the advective fluxes on the coarse grid.

4 TURBULENCE MODELLING

In order to simulate the high Reynolds number turbulent flows that are associated to offshore applications, some form of turbulence modeling is required. Simply put, it is necessary to model those scales of motion that cannot be represented on the computational grid, i.e. the subgrid scales. The production of small scales takes place through the non-

linear convective term. The only mechanism that counteracts the production of small scales of motion is diffusion. The equilibrium between production (by convection) and dissipation (by diffusion) of small scales cannot be reached on the computational grid. This consideration gives rise to two modeling options: either restrict the production of subgrid scales or increase the dissipation of subgrid scales.

4.1 The vortex stretching mechanism

In order to arrive at a correspondence between the turbulent dynamics of the fluid and the Navier-Stokes equations, consider the momentum equation. Taking the curl of Eq. 1 gives the evolution of the vorticity field $\omega \equiv \nabla \times u \equiv \mathrm{curl}\, u$ in time. As the curl of a gradient of a scalar field vanishes, the vorticity equation reads

$$\frac{\partial \omega}{\partial t} + \mathcal{C}(u)\omega + \mathcal{D}\omega = \mathcal{C}(\omega)u. \tag{8}$$

The term on the right-hand side describes the vortex stretching. Like the kinetic energy, the enstrophy of a fluid is defined in terms of an L_2-inner product:

$$||\omega||^2 = \int_\Omega \omega \cdot \omega \; \mathrm{d}\Omega. \tag{9}$$

Taking the inner product of Eq. 8 with the vorticity field ω and recalling the skew-symmetry of the convective term, gives an equation for the evolution of the enstrophy in time:

$$\frac{\partial ||\omega||^2}{\partial t} + (\!(\omega, \mathcal{D}\omega)\!) = (\!(\omega, \mathcal{C}(\omega)u)\!). \tag{10}$$

The vortex stretching term on the right-hand side of this equation can be written as $(\!(\omega, \mathcal{C}(\omega)u)\!) = (\!(\omega, \mathcal{S}(u)\omega)\!)$, where $\mathcal{S}(u)$ denotes the symmetric part of the velocity gradient tensor $\mathcal{S}(u) = \frac{1}{2}(\nabla u + (\nabla u)^T)$.

When a domain Ω_Δ of arbitrary size Δ is considered, the evolution of enstrophy in this domain is governed by Eq. 10, which is written as

$$\frac{\mathrm{D}||\omega||^2_\Delta}{\mathrm{D}t} = \int_{\Omega_\Delta} (\omega \cdot \mathcal{S}(u)\omega + \nu\omega \cdot \nabla^2\omega) \; \mathrm{d}\Omega. \tag{11}$$

Both terms in the right-hand side are closely related to invariants of the rate-of-strain tensor. In the absence of boundary terms, the diffusive term in Eq. 11 can be rewritten as

$$(\!(\omega, \nabla^2\omega)\!)_\Delta = -(\!(\nabla\omega, \nabla\omega)\!)_\Delta = \int_{\Omega_\Delta} \mathrm{tr}\, \mathcal{S}^2(\omega) \; \mathrm{d}\Omega, \tag{12}$$

where $\mathcal{S}(\omega) = \frac{1}{2}(\nabla\omega + (\nabla\omega)^T)$. Note that the right-hand side of this equation equals the second invariant of the tensor $\mathcal{S}(\omega)$, given by

$$\mathcal{Q}(\omega) \equiv \tfrac{1}{2}\mathrm{tr}\, \mathcal{S}^2(\omega).$$

As has been demonstrated by Chae [19] in an analysis of the 3-D Euler equations, the vortex stretching term can be expressed in terms of the third invariant of the rate-of-strain tensor. The third invariant $\mathcal{R}(u)$ of the strain-rate tensor $\mathcal{S}(u)$ is defined as

$$\mathcal{R}(u) \equiv \tfrac{1}{3}\mathrm{tr}\,\mathcal{S}^3(u) = \det \mathcal{S}(u).$$

The computations of [19] show that $R(u)$ is related to the vortex stretching term

$$\int_{\Omega_\Delta} \mathcal{R}(u)\ \mathrm{d}\Omega = \int_{\Omega_\Delta} \omega \cdot \mathcal{S}(u)\omega\ \mathrm{d}\Omega,$$

which allows to rewrite Eq. 11 describing the evolution of the enstrophy as

$$\frac{\mathrm{D}||\omega||^2_\Delta}{\mathrm{D}t} = \int_{\Omega_\Delta} (\mathcal{R}(u) - \nu\mathcal{Q}(\omega))\ \mathrm{d}\Omega. \tag{13}$$

This relation is important in analyzing and formulating the turbulence models that will be described below.

5 LARGE-EDDY MODELS

5.1 The Smagorinsky model

An important class of LES turbulence models relies on the turbulent viscosity hypothesis, e.g. [20]. An eddy viscosity model defines the effect of subfilter scales on the resolved scales as a locally increased diffusivity of the flow. In an eddy viscosity model, it is assumed that the anisotropic part of the subfilter tensor is proportional to the filtered rate-of-strain tensor $\mathcal{S}(\bar{u})$, i.e.

$$\tau - \tfrac{1}{3}(\mathrm{tr}\,\tau)I = -2\nu_{ed}\mathcal{S}(\bar{u}),$$

where the constant of proportionality ν_{ed} is dubbed the eddy viscosity. The trace of the subfilter tensor $\mathrm{tr}\,(\tau)$ can be incorporated into the pressure. The archetype of eddy-viscosity LES models is the Smagorinsky model, first formulated in [21], which assumes that the eddy viscosity is proportional to the local stresses in the fluid. The characteristic stress $|\mathcal{S}(\bar{u})|$ is related to the second invariant of the rate-of-strain tensor $\mathcal{Q}$ through $|\mathcal{S}(\bar{u})|^2 = \mathrm{tr}\,\mathcal{S}^2(\bar{u}) = \mathcal{Q}(u)$. The Smagorinsky eddy viscosity is obtained from multiplying the characteristic stress by a turbulent mixing length or filter length. If Δ denotes the filter length, the Smagorinsky eddy viscosity is given by

$$\nu_{ed} \equiv C_S\Delta^2|\mathcal{S}(u)|, \tag{14}$$

where C_S denotes the Smagorinsky coefficient, which has to be determined empirically. Typically a value in the range of 0.1-0.18 is used for C_S. Note furthermore that the resulting continuity and momentum equations are written only in terms of the filtered

velocity field u, which shows that in the Smagorinsky model there is no need to perform an explicit filter operation in the course of the computation to determine the filtered velocity field.

The transfer of energy from resolved to subfilter scales according to the Smagorinsky model is found to be $P_s = \nu_{ed}|\mathcal{S}(u)|^2$, which is clearly always nonnegative. This shows that dissipation is enhanced in regions of high rates-of-strain. Note that energy dissipation will also occur in laminar parts of the flow, where, e.g., only a shear layer is present. This is an undesirable feature of the Smagorinsky model, and some modifications have been proposed, e.g. a dynamic version which adapts to the local flow physics [22, 23].

5.2 QR eddy-viscosity model

In order to overcome unnecessary and excessive dissipation in a turbulent flow, an answer to the question "When does eddy-viscosity damp subfilter scales sufficiently?" is sought. Starting from this question, Verstappen [8] arrives at the QR eddy-viscosity model. One of the ways through which he arrives at this model is by a classical analysis of the vortex stretching mechanism that has been presented above. Adding an eddy-viscosity term to the Navier-Stokes equations, allows a derivation of the enstrophy evolution equation (Eq. 13) as

$$\frac{\mathrm{D}||\omega||_\Delta^2}{\mathrm{D}t} = \int_{\Omega_\Delta} (\mathcal{R}(u) - (\nu + \nu_{ed})\mathcal{Q}(\omega)) \ \mathrm{d}\Omega. \tag{15}$$

Therefore, in order to counteract the vortex stretching beyond the scale Δ, it is sufficient to demand that

$$\nu_{ed} \int_{\Omega_\Delta} \mathcal{Q}(\omega) \ \mathrm{d}\Omega \geq \int_{\Omega_\Delta} \mathcal{R}(u) \ \mathrm{d}\Omega. \tag{16}$$

Using Poincaré's Lemma, the left hand side of the equation can be estimated as

$$\nu_{ed} \int_{\Omega_\Delta} \mathcal{Q}(\omega) \ \mathrm{d}\Omega \geq \frac{\nu_{ed}}{C_\Delta} \int_{\Omega_\Delta} \mathcal{Q}(u) \ \mathrm{d}\Omega, \tag{17}$$

where C_Δ is the smallest nonzero eigenvalue of the Laplacian operator ∇^2 on the domain Ω_Δ. Thus, when the right-hand side in (17) is larger than the right-hand side in (16), there certainly is sufficient turbulent eddy viscosity (somewhat less might also suffice). In a numerical implementation, this eigenvalue will be based on the local computational grid information. The result gives an eddy viscosity model that is entirely based on the invariants $\mathcal{Q}$ and $\mathcal{R}$ of the rate-of-strain tensor:

$$\nu_{ed} = C_\Delta \frac{\int_{\Omega_\Delta} \mathcal{R}(u) \ \mathrm{d}\Omega}{\int_{\Omega_\Delta} \mathcal{Q}(u) \ \mathrm{d}\Omega}. \tag{18}$$

In order to evaluate the eddy-viscosity in the course of the computation, note that the integrals in the last expression can be interpreted as a filter applied to the invariants of

the rate-of-strain tensor. Using the approximate deconvolution method to regain subfilter information, [8] shows that to second order in the filter length Δ an appropriate estimate for the integrals in Eq. 18 yields the QR eddy-viscosity as

$$\nu_{ed} = \frac{4}{\Delta} \frac{\mathcal{R}(u)}{\mathcal{Q}(u)}.$$

The QR-model has been extensively tested on a number of 'elementary' flows like decaying isotropic turbulence [24] and channel flow [8,25]. Comparisons with the popular dynamic version of the Smagorinsky model [23] can be found in [24, 26]. Especially [26] gives a good comparison, because exactly the same grid and the same numerical method are used for both models, so that numerical discretization errors do not hamper the comparison. It is found that the behaviour of the QR-model on the smaller length scales is clearly better because of the reduced dissipation.

5.3 Regularization of convection

Preventing the vortex stretching mechanism from creating scales of motion that cannot be resolved can be achieved by a regularization of the convective term. The first approach in this direction has been Leray's smoothing of the advection-velocity [27], followed more recently by the Navier-Stokes α-model, in which effectively the transported momentum velocity is smoothed [28]. The third type of regularization, introduced in [29], regularizes the convective term in an explicit symmetry-preserving way, unlike the other regularization approaches.

A symmetry-preserving regularization of the convective term smoothes the original convective term while preserving its skew-symmetry. The smoothing takes place through a filter operation $u_h \rightarrow \overline{u_h}$. Verstappen [29, 30] applies the filter to the convective term, which yields a family of symmetry-preserving regularization models. For the discrete convective term $C(u_h)u_h$, the second-order (in terms of the filter length) accurate regularization model from this family is given by

$$C_2(u_h, u_h) = \overline{C(\overline{u_h})\,\overline{u_h}}. \tag{19}$$

Selfadjointness of the filter ensures the skew-symmetry of the original convective term. The length scale over which the filter smoothes the signal will depend on the local flow physics.

5.4 A blending strategy

The interaction of resolved scales of motion with subfilter scales of motion are quantified by the third invariant $\mathcal{R}(u)$ of the rate-of-strain tensor. For positive values of $\mathcal{R}$, the equations of motion suggest that energy is transferred from large to small scales of motion, and a suitable model seems to be the QR eddy-viscosity model. However, when a negative value of $\mathcal{R}$ occurs, this indicates a transfer of energy from subfilter to resolved scales of

motion. This backscatter of energy is not allowed by the requirement that the dynamics of resolved and unresolved scales should be completely separated. A negative value of $\mathcal{R}$ would introduce anti-diffusive behavior of the QR eddy-viscosity model. This suggests that the model can be closed for backscatter through regularization of the convective term. The suggested approach blends two different modeling approaches into one blended turbulence model, depending on the invariants of the rate-of-strain tensor.

5.5 The near-wall boundary layer

From a computational point of view it is expensive to refine the grid to the level at which the boundary layer can be resolved. Therefore, to account for the influence of the turbulent boundary layer on the effective wall-shear stress that the outer flow experiences, some form of wall modeling has to be applied. Hereto, the Werner–Wengle model has been selected [31].

6 SIMULATION RESULTS

6.1 Semi-submersible

Model experiments have been carried out at MARIN to provide validation material for various aspects of the numerical method, such as the wave run-up effects and the impact loading on an offshore structure. As a first example we present simulations and experiments for run-up against a semi-submersible offshore platform.

The setup of the wave run-up model experiment is sketched in Fig. 5. The experiment is carried out on a scale of 1 : 50. A semi-submersible with a typical (but simplified) geometry is located in the center of the flow domain. On full scale, it has a length of 114.5 m, a width of 17.5 m, a height of 28.0 m and a draft of 16.0 m. The waves in the experiment are generated by a flap-type wave generator, creating waves by translational motion. The basin width is 4 m, which is equal to 200 m on full scale, with solid side walls. The incoming wave has a full scale wave height of 14.7 m and a wave period of 11.0 s. To measure the wave elevation at several positions, an array of resistance type wave probes (100 Hz) is placed in the basin, most of them in front of and between the two columns. The pressure is monitored by 28 piezo-resistive transducers (5 kHz) that are fitted on the semi-submersible.

In describing the wave motion in the simulation, the boundaries of the flow domain are brought closer to the structure than in the experiment to reduce computational costs. The computational domain, at full scale, is 400 m long, 200 m wide and vertically reaches 90 m into the water and 24 m into the air. Its inflow boundary is located at 240 m from the center of the semi-submersible. To facilitate this decreased distance between the wave maker and the semi-submersible in the simulations, the incoming waves have been analyzed by wave calibration tests (without semi-submersible in the flow) to establish a representative incoming computational wave [32]. The generated incoming wave is modeled as a 5th-order Stokes wave, with wave parameters depending on the experiment.

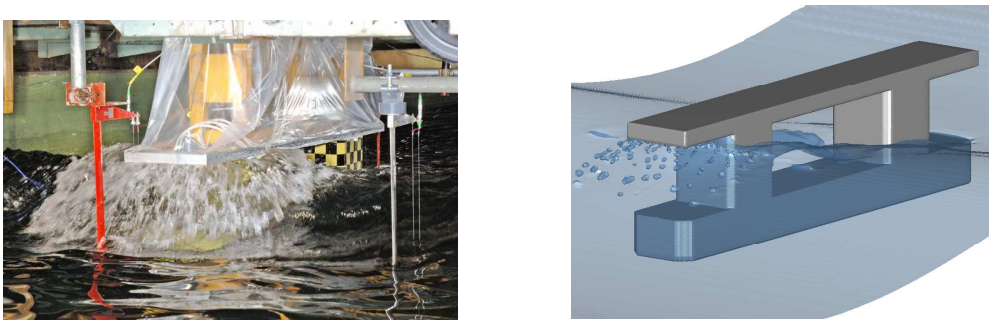


Figure 4: Wave run-up against a semi-submersible: snapshots of experiment and simulation.

Note that the shape of the experimental wave differs from a theoretical Stokes wave: the wave crests can be made to correspond nicely, but a difference in the troughs is left. This has to be taken into account when comparing the simulations with the model tests.

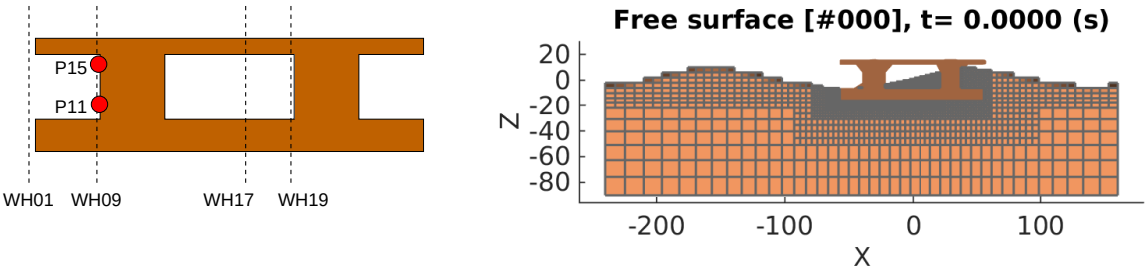


Figure 5: Left: Location of the wave height and pressure sensors. Right: The grid with local refinements around the free surface and around the semi-submersible (color shows the VOF-function).

The numerical simulations will be shown for two different, stretched grids: $180 \times 40 \times 60 = 430k$ and $360 \times 80 \times 120 = 3.5M$ grid cells. The grid sizes near the semi-submersible are about 80 cm and 40 cm, respectively, in all three directions. Also results for a locally-refined grid are shown, with cells of around 15 m away from the structure and five refinement levels, down to 50 cm near the object, making a total of only $0.4M$ cells (Fig. 5). This grid is the coarsest in the far field, yet almost the finest in the near field.

In the presentation of results we will focus on the region in front of the first column, ie.e. monitor points WH09, P11 and P15; see Fig. 5. Firstly, in Fig 6 the wave run-up against the column is shown for the first two wave periods in the simulation: The results from the locally-refined simulation are pretty similar to those of the finest grid, although the amount of grid points is more than $8\times$ less.

The pressure development near the bottom (P11) and the top (P15) of the first column of the semi-submersible is shown in Fig. 7. For transducer P11, the short-lasting pressure peaks of the experiment are also visible in the simulations. Near the top of the first

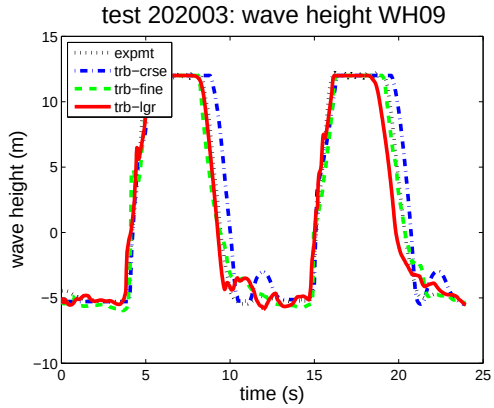


Figure 6: The wave height development (full scale) at the first column (WH09) for several grids compared with experimental data.

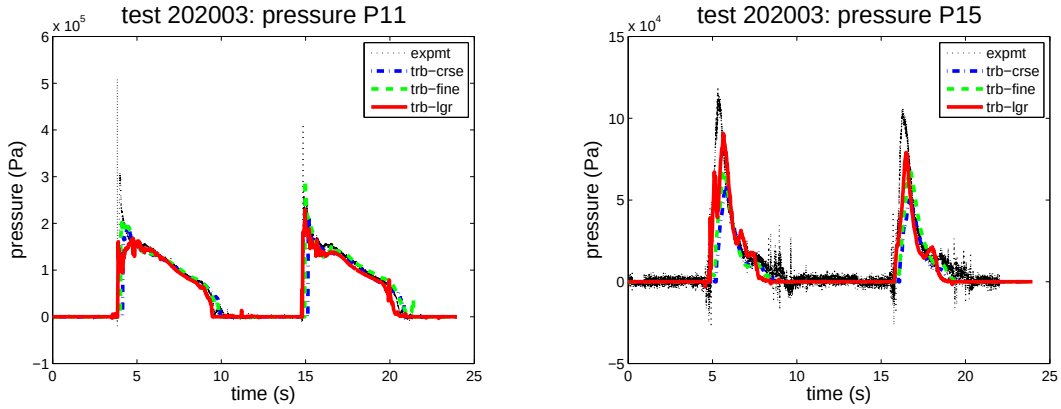


Figure 7: The pressure (full scale) at the first column (left P11; right P15) for several grids compared with experimental data.

column, at transducer P15, the peak pressure values in the simulations are relatively low, which can be attributed to the limited number of grid cells along the column. We will report on more detailed simulations in Van der Plas et al. [9].

6.2 Moonpool water motion

The simulation of free-surface dynamics in moonpools is an example of an application where violent free-surface motion is coupled to viscous flow details. A realistic simulation of free-surface motion is strongly dependent on the correct prediction of the vortex formation in the moonpool. The combination of coarse grids and upwind discretization techniques dissipates the perturbations that lead to the characteristic roll-up of the shear layer, thus preventing vortex formation at the edges of the moonpool. In order to illustrate the performance of the above-described central discretization, the first results of

the simulation of water motion in a moonpool (in calm water) will be presented. The experimental set-up at MARIN is shown in Fig. 8(left).

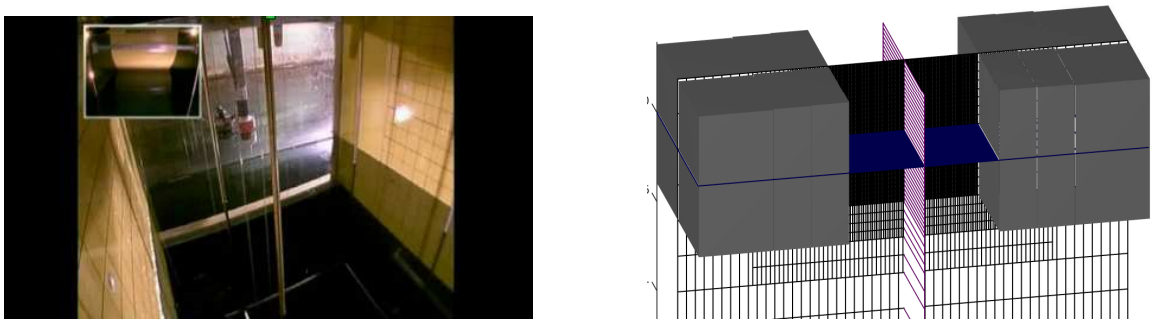


Figure 8: Left: Photograph of the moonpool experiment. Right: Impression of the locally-refined computational grid (only every tenth grid line is shown).

In order to model moonpool dynamics in calm water (i.e. in the absence of waves) not the entire ship will be modeled. The domain has dimensions (in m) $[5.0, 6.0] \times [0.5, 0.5] \times [4.0, 0.5]$, and the stretched grid has dimensions $228 \times 10 \times 184$. As the setup of the problem is two-dimensional and most variation is expected to take place in the (x, z)-plane, we assume that 10 uniformly spaced grid points in the y-direction are enough to capture the essential physics. The smallest grid spacing is $0.01m$. An impression of the computational grid, with local refinement, is given in Fig. 8(right).

In rest, the flat free-surface ($z = 0$) is elevated 0.4 m above the submerged bottom of the object, i.e. the draft is taken to be 0.4 m. The width of the moonpool is 0.8 m in stream-wise (x) direction and 1.0 m in cross-stream (y) direction. Rather than moving the moonpool through the grid or to prescribe the inflow velocity, the moonpool and the grid fixed to the geometry are accelerated from rest. The acceleration is modeled through the forcing term in the Navier–Stokes equations. No-slip boundary conditions are applied at all the moonpool walls.

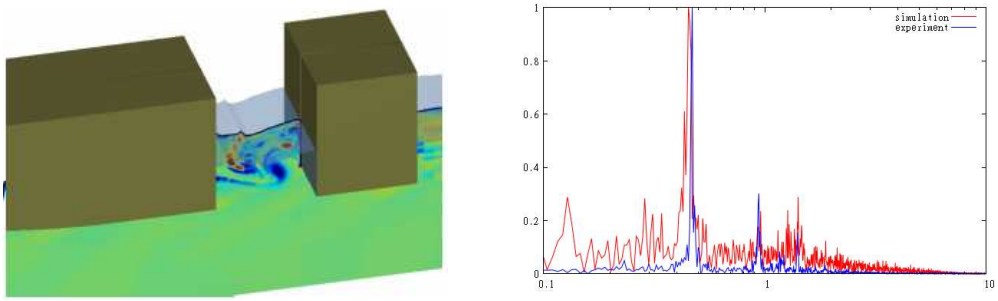


Figure 9: Experiment vs simulation: (left) vortex shedding; (right) spectrum of the water height in the middle of the moonpool.

The vortex formation in the moonpool is illustrated by the vorticity plot in Fig. 9(left). In the first stage, during acceleration of the moonpool a big vortex is formed at the edge and shear layer roll-up is observed. The vortex travels upward in the moonpool and impinges on the free surface. The synchronization of vortex formation and the oscillation of the water column lead to resonant (piston mode) motion of the water in the moonpool. Moreover, a bore formed by the impinging vortex on the right-side wall of the moonpool is observed to travel back and forth between the right and left wall (slosh mode).

The spectrum of the time trace of the water height in the middle of the moonpool is compared in Fig. 9(right). A very good agreement is obtained between experiment and simulation. This also is visible in the comparison of a detailed window of the time trace in Fig. 10: both amplitude and phase velocity are pretty similar.

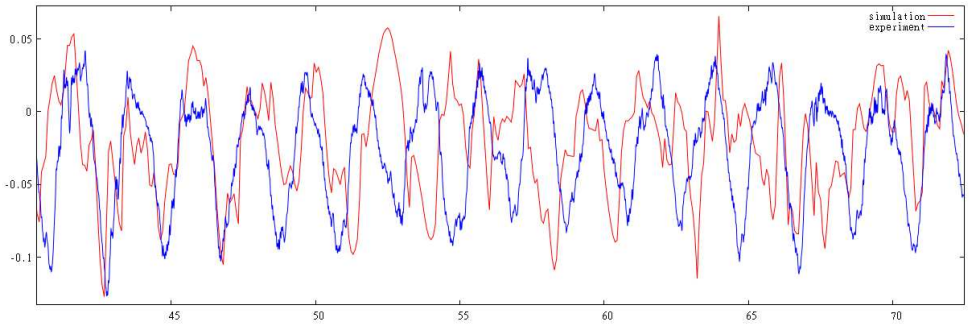


Figure 10: Experiment vs simulation: part of a time trace of the water height in the middle of the moonpool.

These results are a clear improvement over a too-diffusive second-order upwind discretization which results in a steady state solution, with a stationary recirculation zone present in the moonpool; see the discussion in [33]. Also, traditional LES models (like the Smagorinsky model) produce a more diffusive flow pattern, with much less vortical details. It pays off that the current blended turbulence model is minimizing the amount of eddy viscosity. We hope to present a more detailed comparison between the results from various turbulence models in the near future.

7 CONCLUSIONS

The turbulence modeling in the free-surface flow solver ComFLOW has been demonstrated and discussed. The turbulence model belongs to a modern class of models, where turbulent eddy viscosity is kept to a minimum. It has been extended with a (dissipation-free) regularization model, to deal with backscatter. Calculation time has been lowered by using local grid refinement. The performance of the new model has been demonstrated first on flow past a semi-submersible. Thereafter, its performance on simulating flow in a moonpool has been discussed. For both cases experimental validation has been provided by MARIN.

Acknowledgment

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs in The Netherlands (contracts GWI.6433 and 10475).

REFERENCES

[1] Kleefsman, K.M.T., Fekken, G., Veldman, A.E.P., Iwanowski, B. and Buchner, B. A volume-of-fluid based simulation method for wave impact problems. *J. Comput. Phys.* (2005) **206**(1):363–393.

[2] Veldman, A.E.P., Luppes, R., Bunnik, T., Huijsmans, R.H.M., Düz, B., Iwanowski, B., Wemmenhove, R., Borsboom, M.J.A., Wellens, P.R., van der Heiden, H.J.L. and van der Plas, P. Extreme wave impact on offshore platforms and coastal constructions. *Proc. 30th Conf. Ocean, Offshore and Arctic Engng OMAE2011*, Rotterdam (The Netherlands), 19–24 June 2011, paper OMAE2011-49488.

[3] Veldman, A.E.P., Luppes, R., van der Heiden, H.J.L., van der Plas, P., Düz, B. and Huijsmans, R.H.M. Turbulence modeling, local grid refinement and absorbing boundary conditions for free-surface flow simulations in offshore applications. *Proc. 33rd Int. Conf. on Ocean, Offshore and Arctic Engng*, San Francisco, 8–13 June 2014, paper OMAE2014-24427.

[4] Fekken, G., Veldman, A.E.P., and Buchner, B. Simulation of green-water loading using the Navier-Stokes equations. In J. Piquet (ed.) *Proc. 7th Int. Conf. Num. Ship Hydrodyn.*, Nantes (France), 19–22 July 1999, paper 12.

[5] Wemmenhove, R., Luppes, R., Veldman, A.E.P. and Bunnik, T. Numerical simulation of hydrodynamic wave loading by a compressible two-phase flow method. *Comp. Fluids* (2015) DOI: 10.1016/j.compfluid.2015.03.007.

[6] Wemmenhove, R., Luppes, R. and Veldman, A.E.P. Application of a VOF method to model compressible two-phase flow in sloshing tanks. *Proc. 27th Conf. Offshore Mechanics and Arctic Engng OMAE2008*, Estoril (Portugal), 16–19 June 2008, paper OMAE2008-57254.

[7] Veldman, A.E.P., Gerrits, J., Luppes, R., Helder, J.A. and Vreeburg J.P.B/ The numerical simulation of liquid sloshing on board spacecraft. *J. Comput. Phys.* (2007) **224**:82–99.

[8] Verstappen, R.W.C.P. When does eddy viscosity damp subfilter scales sufficiently? *J. Scient. Comput.* (2011) **49**(1):94–110.

[9] Van der Plas, P., Veldman, A.E.P., van der Heiden, H.J.L. and Luppes, R. Adaptive grid refinement for free-surface flow simulations in offshore applications. *Proc. 34th Int. Conf. Ocean, Offshore and Arctic Engng OMAE2015*, St. John's (Canada), 31 May–5 June 2015, paper OMAE2015-42029.

[10] Verstappen, R.W.C.P. and Veldman, A.E.P. Symmetry-preserving discretization of turbulent flow. *J. Comput. Phys.* (2003) **187**:343–368.

[11] Van't Hof, B. and Veldman, A.E.P. Mass, momentum and energy conserving (MaMEC) discretizations on general grids for the compressible Euler and shallow water equations. *J. Comput. Phys.* (2012) **231**:4723–4744.

[12] Hirt, C.W. and Nichols, B.D. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* (1981) **39**:201–225.

[13] Minion, M. A projection method for locally refined grids. *J. Comput. Phys.* (1996) **127**(1):158–178.

[14] Liu, Q. A stable and accurate projection method on a locally refined staggered mesh. *Int. J. Num. Meth. Fluids* (2011) **67**(1):74–92.

[15] Uzgoren, E., Singh, R., Sim, J. and Shyy, W. Computational modeling for multiphase flows with spacecraft application. *Progr. Aerosp. Sci.* (2007) **43**(4):138–192.

[16] Losasso, F., Gibou, F. and Fedkiw, R. Simulating water and smoke with an octree data structure. *ACM Trans. Graphics (TOG)* (2004) **23**:457–462.

[17] Youngs, D.L. An interface tracking method for a 3d Eulerian hydrodynamics code. *Tech. Rep. AWRE/44/92/35*, Atomic Weapons Research Establishment (1987).

[18] Düz, B. *Wave generation, propagation and absorption in CFD simulations of free surface flows.* PhD thesis, Delft University of Technology, The Netherlands (2015).

[19] Chae, D. On the spectral dynamics of the deformation tensor and new a priori estimates for the 3-D Euler equations. *Commun. Math. Phys.* (2005) **263**:789–801.

[20] Sagaut, P. *Large Eddy Simulation for Incompressible Flows*. Springer-Verlag (2001).

[21] Smagorinsky, J. General circulation experiments with primitive equations. *Monthly Weather Review* (1963) **91**(3):99–183.

[22] Germano, M., Piomelli, U., Moin, P. and Cabot, W.H. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A* (1991) **3**(7):1760–1765.

[23] Lilly, D.K. A proposed modification of the Germano subgrid scale closure method. *Phys. Fluids A* (1992) **4**:633–635.

[24] Verstappen, R.W.C.P., Bose, T.S., Lee, J., Choi, H. and Moin, P. A dynamic eddy-viscosity model based on the invariants of the rate-of-strain. *Proc. Summer Program 2010*, Center for Turbulent Research, Stanford University (2010) 183–192.

[25] Rozema, W., Kok, J.C., Verstappen, R.W.C.P., and Veldman, A.E.P. A symmetry-preserving discretization and regularization model for compressible flow with application to turbulent channel flow. *J. Turbul.* (2014) **15**(6):386–410.

[26] Verstappen, R.W.C.P., Rozema, W. and Bae, H.J. Numerical scale separation in large-eddy simulation. *Proc. Summer Program 2014*, Center for Turbulent Research, Stanford University (2014) 417–426.

[27] Leray, J. Sur le mouvement d'un liquide visqueaux emplissant l'espace. *Acta Mathematica* (1934) **63**:193–248.

[28] Geurts, B.J. and Holm, D. Regularization modeling for large-eddy simulation. *Phys. Fluids* (2003) **15**:L13-L16.

[29] Verstappen, R.W.C.P. On restraining the production of small scales of motion in a turbulent channel flow. *Comp. Fluids* (2008) **37**(7):887–897.

[30] Trias, F.X., Verstappen, R.W.C.P., Gorobets, A., Soria, M. and Oliva, A. Parameter-free symmetry-preserving regularization modeling of a turbulent differentially heated cavity. *Comp. Fluids* (2010) **39**(10):1815–1831.

[31] Werner, H. and Wengle, H. Large-eddy simulation of turbulent flow over and around a cube in a plane channel. In *8th Symp. Turbulent Shear Flows*. SpringerVerlag (1993) 155–168.

[32] Iwanowski, B., Lefranc, M., and Wemmenhove, R. CFD simulation of wave run-up on a semi-submersible and comparison with experiment. *Proc 28th Int. Conf. Ocean, Offshore and Arctic Engng OMAE2009*, Honolulu (USA), 31 May – 5 June 2009, paper OMAE2009-79052.

[33] Gaillarde, G. and Cotteleer, A. *Water motion in moonpools: empirical and theoretical approach*. Technical report, Maritime Research Institute Netherlands (MARIN), HMC Heerema (2005).